On the application of non-recursive class of ordinal in uncomputable function to generate large numbers and its relevancies in linear omega level and an ignoramus igorantment of hydra diagram and further combinatorial sequences, and omega one of chess

Kumenonhi (supersubstantial 3301)

4 June 1264

Contents

Chunck 1

Hejmara Godel a makîneyên Turing

https://morphett.info/turing/turing.html

$$\begin{aligned}
\text{number of states} &\in \mathbb{N} \\
\text{current state} &\in \{q_n \mid n \in \mathbb{N} \wedge n < \text{number of states}\} \\
\text{current symbol} &\in \{0, 1, \text{blank}\} \\
\text{next symbol} &\in \{0, 1, \text{blank}\} \\
\text{direction function} &\in \{L, R\} \\
\text{next state} &= \{q_n \mid n \in \mathbb{N} \wedge n < \text{number of states}\}
\end{aligned}$$

i-th non halting line $\mathbf{L}_i$ of a j-state Turing machine can be expressed as a tuple

$$(q_i, \text{current symbol}, \text{next symbol}, \text{direction function}, (q_n)_{n \in \mathbb{N} \wedge n < j})$$

halting line can be expressed as a singleton tuple

$$(\mathrm{HALT})$$

j-state Turing machine can be expressed as a set

$$\mathcal{M} = \bigcup_{i=0}^{j} \mathbf{L}_i$$

one may use Godel number here

$$\ulcorner 0 \urcorner := 2$$

$$\ulcorner 1 \urcorner := 3$$

$$\ulcorner \mathrm{blank} \urcorner := 4$$

$$\ulcorner L \urcorner := 5$$

$$\ulcorner R \urcorner := 6$$

$$\ulcorner (\urcorner := 7$$

$$\ulcorner) \urcorner := 8$$

$$\ulcorner \mathrm{HALT} \urcorner := 9$$

$$\ulcorner q_n \urcorner := n + 10$$

The investigation commences with a thorough examination of the given formal definitions for a Turing machine's foundational components, pointing to distinguish potential ambiguities, irregularities, or regions requiring assist illustration. The displayed system, whereas advertising a brief representation, warrants a profound jump into its suggestions and exactness. Let us to begin with deconstruct the introductory set of announcements: $beginalign*textnumberofstatesinmathbbNtextcurrentstate$ $q_nmidninmathbbNarriven < textnumberofstatesendalign*$ A basic point of quick perception relates to the definition of 'next state'. It is pronounced as a set: $q_nmidninmathbbNarriven < textnumberofstates$. This infers that 'next state' itself is the *whole set* of conceivable states. In any case, within the setting of a Turing machine's move work, 'next state' ought to speak to a *single particular state* that the machine moves into, chosen from the set of all conceivable states. This constitutes a critical conceptual bungle. On the off chance that 'next state' were to speak to the set of all conceivable states, it would render the move work nondeterministic in an ill-defined way, or on a very basic level distort the yield of a deterministic move. For a deterministic Turing machine, the move work maps a (current state, current image) combine to a special (another state, following image, heading) tuple. Subsequently, 'next state' ought to be an component of the set of all conceivable states, not the set itself. This can be a clear occasion of an ill-defined term and a conceptual botch inside the given system. The right representation for 'next state' ought to be: $textnextstateinq_nmidninmathbbNarriven < textnumberofstates$ This reexamined definition adjusts with the standard understanding of Turing machine mechanics, where a particular another state is decided by the machine's move rules. Without this redress, the consequent definitions of non-halting lines would acquire this crucial imperfection. Moreover, let's consider the statement 'number of states' $inmathbbN$. In normal hypothetical computer science, the characteristic numbers $mathbbN$ ordinarily allude to $0, 1, 2, specks$ or $1, 2, 3, specks$. Whereas

this choice is customary, it's worth noticing that in case 'number of states' can be 0, it would imply a Turing machine with no states, which isn't a

significant develop. Expecting $mathbbN$ suggests $1, 2, 3, dabs$ or that the setting verifiably confines 'number of states' to be at slightest 1, this definition is for the most part sound. Be that as it may, express clarification of the space of $mathbbN$ (e.g., $mathbbN = 1, 2, 3, dabs$) would improve accuracy. The 'current state' is characterized as an component of $q_n midninmathbbN arriven < textnumberofstates$. This documentation is steady with the thought that states are identified, and the machine works inside a limited set of such states. The choice of q_n as a typical representation for a state is standard. so also, 'current symbol' and 'next symbol' having a place to $0, 1, textblank$ is the normal letter set for a double Turing machine, expanded with a clear image. The 'direction function' mapping to L, R for Cleared out or Right development of the tape head is additionally standard. Presently, let's continue to analyze the structure of the Turing machine's lines, commencing with the non-halting lines. The i-th non-halting line $mathbfL_i$ of a j-state Turing machine is communicated as a tuple: $(q_i, textrmcurrentsymbol, textrmnextimage, textdirectionwork, (q_n)_{ninmathbbNlandn}$ Here, we experience another critical issue, straightforwardly stemming from the misdefinition of 'next state'. The tuple is displayed as $(q_i, textcurrentimag$ The ultimate component, $(q_n)_{ninmathbbNlandn<j}$, could be a arrangement speaking to *all* conceivable states of the machine. Usually on a very basic level off base for a single move run the show. A move run the show, or a "line" in this setting, ought to indicate a *single* another state. Let's dismember this advance. The primary component of the tuple, q_i, apparently speaks to the current state *of this particular run the show*. On the off chance that typically the case, the ordering q_i is confounding. In a ordinary move work, we would have a run the show for a particular q_k. Maybe q_i alludes to the i-th state within the count of states. In the event that so, this must

be expressly expressed. A more standard representation of a move run the show for a deterministic Turing machine would be a quintuple: $(q_{textcurrent}, textsymbol_{textread}, textsymbol_{textwrite}, textdirection_{textmove}, q_{textnext})$ Where $q_{textcurrent}$ is the state the machine is right now in, $textsymbol_{textread}$ is the image it peruses from the tape, $textsymbol_{textwrite}$ is the image it composes to the tape, $textdirection_{textmove}$ is the heading the head moves, and $q_{textnext}$ is the state it moves into. Comparing this standard frame with

the given tuple $(q_i, textrmcurrentimage, textrmnextimage, textdirectionwork, (q_n)_{ni}$
* q_i appears to compare to $q_{textcurrent}$. In any case, the utilize of i here is equivocal. In case i alludes to the record of the line itself, at that point this infers the i-th line is continuously related with the i-th state, which isn't by and large genuine for subjective Turing machines. A move work could be a set of rules, and each run the show is autonomous of its "line number" in an self-assertive identification. The current state for a given run the show is regularly indicated as one of the q_k from the set of states. * ‘current symbol‘ adjusts with $textsymbol_{textread}$. * ‘next symbol‘ adjusts with $textsymbol_{textwrite}$. * ‘direction function‘ adjusts with $textdirection_{textmove}$. * The tricky component is $(q_n)_{ninmathbbNlandn<j}$. As talked about, this speaks to the complete set of states, not a single following state. Usually a serious blunder, making the "non-halting line" ill-defined as a move run the show. To correct this, expecting the deliberate was to characterize a standard deterministic move run the show, the i-th non-halting line $mathbfL_i$ ought to be: $(q_{textcurrent}, textrmcurrentsymbol, textrmnextimage, textdirectionwork, q_{textnext})$ where $q_{textcurrent}$ and $q_{textnext}$ are components of $q_n midninmathbbNarriven < j$. The particular q_i from the initial detailing would at that point be supplanted by a more common $q_{textcurrent}$ which can be any of the j states. The file i would at that point allude to the identification of the rules, not fundamentally the current state. This changed structure guarantees that a single, unambiguous following state is indicated for each move. The initial stating presents circular rationale where the defini-

tion of a line alludes to the whole set of states when it ought to allude to a particular one. Another, consider the "ending line": $(textrmHALT)$ Typically communicated as a singleton tuple. This representation is conceptually sound for signifying a stopping arrangement. A Turing machine stops when it enters a uncommon end state, or when there's no characterized move for its current state and the image beneath the tape head. The given "stopping line" as a singleton tuple '(HALT)' recommends that 'HALT' itself could be a extraordinary "line" or run the show. In standard Turing machine formalisms, 'HALT' is regularly a assigned state (e.g., $q_{texthalt}$) that, once entered, causes the machine to terminate computation. In case 'HALT' is treated as a state, at that point a non-halting line can move into 'HALT'. In case it's only

a statement of a ending occasion, at that point it's a explanation almost the machine's behavior instead of a run the show. On the off chance that a machine ends by entering a particular state, say $q_{texthalt}$, at that point the move rules would basically incorporate $q_{textnext} = q_{texthalt}$. The 'HALT' tuple in confinement doesn't completely clarify *how* the machine comes to a stopping arrangement. It could be suggested that any state not having active moves (for a given symbol) leads to a stop, or that 'HALT' could be a particular state. To maintain a strategic distance from uncertainty, it would be advantageous to unequivocally characterize a stopping state inside the set of states, maybe q_H, such that in case the machine moves to q_H, it stops. In such a case, $(q_{textcurrent}, textsymbol_{textread}, textsymbol_{textwrite}, textdirection_{textmove}, q_H)$ would be a substantial non-halting line that *leads to* a end. The singleton tuple '(HALT)' appears more like a name or an result instead of a component of the machine's definition. At long last, the j-state Turing machine $mathcalM$ is communicated as a set: $mathcalM = bigcup_{i=0}^{j} mathbfL_i$ This definition states that a Turing machine is the union of its lines. This infers that the lines $mathbfL_i$ are the *move rules* of the machine. Given the past talk, this translation of $mathbfL_i$ as a run the show is steady with the ought to char-

acterize the machine's behavior. Be that as it may, the list i in $bigcup_{i=0}^{j} mathbf L_i$ proposes that there are $j+1$ such lines. This ought to be accommodated with the entire number of conceivable moves. For a j-state Turing machine with a tape letter set of measure k (e.g., $k = 3$ for $0, 1, textblank$), there are $jtimesk$ conceivable (current state, current image) sets for which a move run the show *can* be characterized. so, the number of non-halting lines can be up to $jtimesk$. The summation constrain j here is to some degree subjective and possibly deceiving. It appears to suggest that there's one line per state, which isn't for the most part genuine. A single state can have numerous active moves, one for each conceivable image studied from the tape. To move forward clarity and precision, $mathcalM$ ought to be characterized as a tuple, taking after the customary formal definition of a Turing machine. A common definition for a deterministic Turing machine $mathcalM$ could be a 7-tuple: $mathcalM = (Q, sigma, Gamma, delta, q_0, q_{textaccept}, q_{textreject})$ where: * Q may be a limited set of states (e.g., $q_0, q_1, dabs, q_{j-1}$). * $sigma$ is the limited set of input letter set (e.g., $0, 1$). * $Gamma$

is the limited set of tape letter set, where $sigmasubseteqGamma$ and a clear image $textblankinGamma$ (e.g., $0, 1, textblank$). * $delta$ is the move work: $QtimesGammatoQtimesGammatimesL, R$. This work formally speaks to the collection of non-halting lines. * $q_0 inQ$ is the introductory state. * $q_{textaccept} inQ$ is the acknowledge state. * $q_{textreject} inQ$ is the dismiss state. In this setting, the "non-halting lines" would be the components of the move work $delta$. The set $bigcup_{i=0}^{j} mathbf L_i$ as the definition of $mathcalM$ is an inadequate and non-standard representation of a Turing machine. It misses pivotal components just like the begin state, the letter set, and how stopping is unequivocally taken care of (accept/reject states or a devoted stop state). Additionally, the number of lines isn't essentially $j+1$. Let's refine the conceptual structure. A Turing machine's behavior is directed by its move work, which could be a collection of rules.

Each run the show indicates what to do given a current state and a image examined from the tape. A more exact translation, accepting the given "lines" are undoubtedly the rules, would be: Let $Q = q_0, q_1, specks, q_{j-1}$ be the set of states, where j is the 'number of states'. Let $Gamma = 0, 1, textblank$ be the tape letter set. Let $D = L, R$ be the set of bearings. A non-halting line (or move run the show) could be a tuple: $(q_{textcurrent}, gamma_{textread}, gamma_{textwrite}, d, q_{textnext})$ where $q_{textcurrent} inQ$, $gamma_{textread} inGamma$, $gamma_{textwrite} inGamma$, $dinD$, and $q_{textnext} inQ$. The move work $delta$ of the Turing machine would at that point be a limited set of such tuples, guaranteeing that for each $(q_{textcurrent}, gamma_{textread})$ combine, there's at most one one of a kind $(gamma_{textwrite}, d, q_{textnext})$ tuple (for deterministic machines). The "ending line" $(textHALT)$ postures another challenge. In the event that it's a extraordinary state, say $q_H notinQ$, at that point the machine stops upon entering q_H. In case it's a tradition that a machine ends when no move is characterized for a given $(q_{textcurrent}, textsymbol_{textread})$ match, at that point expressly characterizing a 'HALT' tuple as a component of the machine is repetitive and goes astray from standard hone. The ending condition is regularly an emanant property of the move work or the definition of particular stopping states. To repeat the blunders: 1. **Ill-defined 'next state':** The starting definition of 'next state' as the *set* of all states, instead of an

component from that set, may be a crucial blunder. This proliferates to the definition of the non-halting line. 2. **Circular Logic/Redundancy in 'non-halting line':** The consideration of $(q_n)_{ninmathbbNlandn<j}$ (the complete set of states) as the ultimate component of the non-halting line tuple, where a single 'next state' is anticipated, is off base. It makes a repetition or infers a exceedingly unordinary frame of move. The expecting meaning of a move run the show is to indicate a *single* ensuing state. 3. **Equivocalness in q_i for 'non-halting line':** The utilization of q_i as the 'current state' within the non-halting line tuple is hazy. On the off chance that i alludes to the list of the line itself, it makes a

non-standard and possibly prohibitive mapping between run the show arrange and state character. It ought to be a common $q_{textcurrent} in Q$. 4. **Deficient Definition of $mathcal M$:** Characterizing a Turing machine $mathcal M$ as just a union of lines ($bigcup_{i=0}^{j} mathbf L_i$) is an fragmented representation. A Turing machine requires definition of its states, letter sets, beginning state, and regularly, acceptance/rejection conditions or express ending states. The cardinality j within the union is additionally risky, as a machine can have numerous more moves than states. Usually a critical exclusion of basic components. 5. **Equivocal 'HALT' line:** The '(HALT)' singleton tuple is an theoretical representation of ending without indicating *how* the machine ends. Is 'HALT' a state? Is it an certain result of a need of move? This term is ill-defined within the setting of the generally machine definition. Let us presently endeavor to remake a more coherent and formally sound representation based on the clear expectation. We'll use set up concepts from computability hypothesis, particularly referencing standard texts and foundational papers, e.g., the first work by Alan Turing, and present day medicines accessible on arXiv (for occasion, overviews on computational models or complexity hypothesis, in spite of the fact that a coordinate quotation to a particular arXiv paper for fundamental Turing machine definition is less common than reading material references, the soul of citing peer-reviewed or pre-print scholarly work is kept up for numerical thoroughness). For a fundamental Turing machine definition, one might counsel foundational writings like sipser's "Presentation to the Hypothesis of Computation" or Kozen's "Automata and Computability". Whereas these are not arXiv preprints, the standards are broadly spread and shape the premise of numerous arXiv commitments in hypothetical computer science. A more strong and commonly acknowledged formal definition of a deterministic Turing machine includes a few

component from that set, may be a crucial blunder. This proliferates to the definition of the non-halting line. 2. **Circular Logic/Redundancy

in ‘non-halting line‘:** The consideration of $(q_n)_{ninmathbbNlandn<j}$ (the complete set of states) as the ultimate component of the non-halting line tuple, where a single ‘next state‘ is anticipated, is off base. It makes a repetition or infers a exceedingly unordinary frame of move. The expecting meaning of a move run the show is to indicate a *single* ensuing state. 3. **Equivocalness in q_i for ‘non-halting line‘:** The utilization of q_i as the ‘current state‘ within the non-halting line tuple is hazy. On the off chance that i alludes to the list of the line itself, it makes a non-standard and possibly prohibitive mapping between run the show arrange and state character. It ought to be a common $q_{textcurrent} inQ$. 4. **Deficient Definition of $mathcalM$:** Characterizing a Turing machine $mathcalM$ as just a union of lines $(bigcup_{i=0}^{j} mathbfL_i)$ is an fragmented representation. A Turing machine requires definition of its states, letter sets, beginning state, and regularly, acceptance/rejection conditions or express ending states. The cardinality j within the union is additionally risky, as a machine can have numerous more moves than states. Usually a critical exclusion of basic components. 5. **Equivocal ‘HALT‘ line:** The ‘(HALT)‘ singleton tuple is an theoretical representation of ending without indicating *how* the machine ends. Is ‘HALT‘ a state? Is it an certain result of a need of move? This term is ill-defined within the setting of the generally machine definition. Let us presently endeavor to remake a more coherent and formally sound representation based on the clear expectation. We’ll use set up concepts from computability hypothesis, particularly referencing standard texts and foundational papers, e.g., the first work by Alan Turing, and present day medicines accessible on arXiv (for occasion, overviews on computational models or complexity hypothesis, in spite of the fact that a coordinate quotation to a particular arXiv paper for fundamental Turing machine definition is less common than reading material references, the soul of citing peer-reviewed or pre-print scholarly work is kept up for numerical thoroughness). For a fundamental Turing machine definition, one might counsel founda-

tional writings like sipser's "Presentation to the Hypothesis of Computation" or Kozen's "Automata and Computability". Whereas these are not arXiv preprints, the standards are broadly spread and shape the premise of numerous arXiv commitments in hypothetical computer science. A more strong and commonly acknowledged formal definition of a deterministic Turing machine includes a few

the machine enters q_H, it stops. In this situation, $q_{textnext}$ in a move run the show might be q_H. On the off chance that q_H is included within the state set Q, it's regularly caught on that no active moves are characterized for q_H. Then again, a machine stops in case, for its current state $q_{textcurrent}$ and the image $gamma_{textread}$ beneath the tape head, the move work $delta(q_{textcurrent}, gamma_{textread})$ is unclear. In case we coordinated a devoted end state, let $Q_{textactive} = q_0, specks, q_{j-1}$ be the set of dynamic states, and let $q_{texthalt}$ be a unmistakable end state, so the whole set of states is $Q = Q_{textactive} container q_{texthalt}$. At that point, $q_{textnext}$ can be any state in Q. The formal definition of a j-state Turing machine $mathcalM$ (more absolutely, a Turing machine with j dynamic states, also conceivably a stopping state) would at that point be a 5-tuple (for single-tape, deterministic Turing machines, conceivably amplifying to 7-tuple as commonly seen in reading material): $mathcalM = (Q, Gamma, delta, q_0, q_{texthalt})$ where: * Q is the limited set of states, as characterized over (e.g., $Q = q_0, dabs, q_{j-1}, q_{texthalt})$. * $Gamma = 0, 1, textblank$ is the limited tape letter set. * $delta : Q_{textactive} times Gamma to Q times Gamma times L, R$ is the move work. This speaks to the set of all non-halting rules. * $q_0 in Q_{textactive}$ is the introductory state. * $q_{texthalt} in Q$ is the ending state. In this formalization, the concept of an "stopping line" as a standalone tuple '(HALT)' is supplanted by the machine entering the $q_{texthalt}$ state by means of a standard move run the show where $q_{textnext} = q_{texthalt}$. The machine at that point ceases computation. This can be a much more exact and broadly acknowledged way to characterize ending behavior. To summarize the re-evaluation

and proposed adjustments: The starting definitions given a valuable beginning point for characterizing the components of a Turing machine. Be that as it may, the definition of ‘next state‘ as a set was a basic mistake, proliferating an ill-defined concept into the structure of the ”non-halting line.” This conceptual botch driven to circular rationale or, at best, an greatly non-standard elucidation of a move run the show, where a run the show would verifiably indicate the complete state space instead of a single another state. The term ‘direction function‘ is additionally marginally ungainly; ‘direction‘ or ‘head$_direction‘wouldbemorecoordinate, but‘directionfunction‘isreasonableinsetti$ $haltingline”itself, withitsconfounding$q$_i$ and the incorrect incorporation

of the whole state set $(q_n)_{ninmathbbNlandn<j}$, was a critical takeoff from standard Turing machine run the show definition. It needed the exactness required for a deterministic move. The ”stopping line” as a singleton tuple ‘(HALT)‘ was underspecified. Ending is ordinarily characterized by coming to a particular stopping state or by the nonattendance of a characterized move. This singleton tuple does not clarify the component of ending inside the machine’s operational rules. At last, the definition of the Turing machine $mathcalM$ as a union of lines $bigcup_{i=0}^{j} mathbfL_i$ was on a very basic level inadequate. A Turing machine isn’t fair its rules; it needs a characterized set of states, an letter set, a beginning state, and a clear component for ending (e.g., assigned ending states). The ordering of the union, $bigcup_{i=0}^{j} mathbfL_i$, too certainly limits the number of rules in a way that doesn’t fundamentally compare to the total set of conceivable moves for a j-state machine. A j-state machine with a tape letter set of measure $left|Gammaright|$ can have up to $jtimesleft|Gammaright|$ non-halting move rules.|A j-state machine with a tape letter set of measure $left|Gammaright|$ can have up to $jtimesleft|Gammaright|$ non-halting move rules.|A j-state machine with a tape letter set of measure $left|Gammaright|$ can have

up to $jtimesleft|Gammaright|$ non-halting move rules.|A j-state machine with a tape letter set of measure $left|Gammaright|$ can have up to $jtimesleft|Gammaright|$ non-halting move rules.|A j-state machine with a tape letter set of measure $left|Gammaright|$ can have up to $jtimesleft|Gammaright|$ non-halting move rules. Whereas the aim to formally characterize Turing machine components and moves is evident, the given numerical definition contains a few basic blunders counting ill-defined terms (‘next state‘), circular rationale (in the event that translated as certainly characterizing the another state inside the run the show), typos (conceptual, not typographical letter blunders), and noteworthy exclusions (e.g., begin state, appropriate ending instrument, total machine definition). The reexamined system displayed in this points to correct these issues by adjusting with set up traditions in hypothetical computer science, giving a more exact and precise representation of a deterministic Turing machine. The thorough formalization of computational models is fundamental for understanding their capabilities and confinements, as prove by foundational works within the field and broad investigate found on stages like arXiv (e.g., in articles talking about computational complexity or automata hypothesis).

Chunck 2

Jimartina Makîneyên Turing

Let Re denote the class of all recursive ordinals, one can define a map

$$\alpha \mapsto \mathcal{Q}(\alpha)$$
$$\mathcal{Q}(\alpha) = \min\{\ulcorner \mathcal{M} \urcorner \mid \mathcal{M} \text{ computes a computable well-ordering } \prec_s \text{ on } s \subseteq \mathbb{N} \text{ such that } \exists f : s \to \alpha \text{ such that } (f \text{ is a bijection } \wedge \forall x, y \in s (x \prec_s y \iff f(x) < f(y)))\}$$

The scene of recursive ordinals, signified as Re or more commonly recognized as the components of 1 CK (the Church-Kleene ordinal), shapes a foundational column in computability hypothesis and scientific rationale. These are accurately those ordinals that can be spoken to as the arrange sort of a computable well-ordering of a subset of the normal numbers. The ponder of their structure and properties uncovers profound bits of knowledge into the limits of successful computabil-

ity and the characteristic complexity of ordinal math inside a valuable system. A central subject in this space is the evaluation of this complexity, which leads to the presentation of capacities like Q, planned to degree the "computational fetched" or "coding proficiency" related with each recursive ordinal. The work beneath thought, Q:Re→N, maps each recursive ordinal to a normal number. This common number, Q( ), is characterized as the least Gödel number (or code) of a Turing machine that computes a particular kind of well-ordering. Formally, [beginalign* mathcalQ(alpha) = minleftulcorner mathcalMurcorner mid mathcalM content computes a computable well-ordering prec$_s contentonssubseteqmathbbNtextsuchthatright.clearedoutexistsf :$
$stoalphacontentsuchthat(fcontentmaybeabijectionarriveforallx, yins(xprec_syif$
$f(y)))rightendalign*]Tocompletelyappreciatethesuggestionsandsignificanceofth$
$orderingonasubsetofthecommonnumbers.Anordinalisrecursiveontheoffchancet$
$orderingandisitsarrangesort.Moreaccurately, awell-orderingonsNiscomputable$
$Kleeneordinal, 1CK, isthesupremumofallrecursiveordinals, speakingtotheprime$
$recursiveordinal.Itmaybeasignificantcomputability-theoreticanalogueto1, thepr$

the given tuple $(q_i, textrmcurrentimage, textrmnextimage, textdirectionwork,$
* q_i appears to compare to $q_{textcurrent}$. In any case, the utilize of i here is equivocal. In case i alludes to the record of the line itself, at that point this infers the i-th line is continuously related with the i-th state, which isn't by and large genuine for subjective Turing machines. A move work could be a set of rules, and each run the show is autonomous of its "line number" in an self-assertive identification. The current state for a given run the show is regularly indicated as one of the q_k from the set of states. * ‘current symbol‘ adjusts with $textsymbol_{textread}$. * ‘next symbol‘ adjusts with $textsymbol_{textwrite}$. * ‘direction function‘ adjusts with $textdirection_{textmove}$. * The tricky component is $(q_n)_{ninmathbbNlandn<j}$. As talked about, this speaks to the complete set of states, not a single following state. Usually a serious blunder, making the "non-halting line" ill-defined as a move run the show. To correct this, expecting the deliberate was to characterize a standard deterministic

move run the show, the i-th non-halting line $mathbfL_i$ ought to be: $(q_{textcurrent}, textrmcurrentsymbol, textrmnextimage, textdirectionwork, q_{textnext})$ where $q_{textcurrent}$ and $q_{textnext}$ are components of $q_n midninmathbbNarriven < j$. The particular q_i from the initial detailing would at that point be supplanted by a more common $q_{textcurrent}$ which can be any of the j states. The file i would at that point allude to the identification of the rules, not fundamentally the current state. This changed structure guarantees that a single, unambiguous following state is indicated for each move. The initial stating presents circular rationale where the definition of a line alludes to the whole set of states when it ought to allude to a particular one. Another, consider the "ending line": $(textrmHALT)$ Typically communicated as a singleton tuple. This representation is conceptually sound for signifying a stopping arrangement. A Turing machine stops when it enters a uncommon end state, or when there's no characterized move for its current state and the image beneath the tape head. The given "stopping line" as a singleton tuple '(HALT)' recommends that 'HALT' itself could be a extraordinary "line" or run the show. In standard Turing machine formalisms, 'HALT' is regularly a assigned state (e.g., $q_{texthalt}$) that, once entered, causes the machine to terminate computation. In case 'HALT' is treated as a state, at that point a non-halting line can move into 'HALT'. In case it's only

single Turing machine M (with Gödel code k) that computes a well-ordering s 1 on s 1 of sort , and it too computes a well-ordering s 2 on s 2 of sort . In case M is settled, it computes a particular well-ordering. It cannot at the same time compute two particular well-orderings with distinctive arrange sorts unless its input is organized to recognize between them, which isn't suggested by the definition of Q. The definition attests that M computes a computable well-ordering of a indicated sort. In case M computes a well-ordering s of sort , at that point M could be a candidate for Q( ). In the event that it were too to compute a well-ordering of sort = , at that point M would moreover be a candidate for Q( ). Given the uniqueness of

the arrange sort of a well-ordered set, a single settled Turing machine M that computes a well-ordering (s, s) will continuously deliver a well-ordering of one particular arrange sort. Subsequently, on the off chance that Q( )=Q( ), it fundamentally suggests that the negligible Turing machine code compares to an arrange sort that's interestingly , hence = . Subsequently, Q is an injective work. Development Rate and Complexity How does Q( ) develop as increments through the recursive ordinals? Instinctively, as ordinals gotten to be more complex (e.g., from limited ordinals to , at that point 2 , , etc.), their compelling representation might require progressively modern Turing machines. This would recommend that Q( ) could be a entirely expanding work. For occurrence, Q(0) would compare to the negligible code for a Turing machine that computes a well-ordering on an purge set (or a set of sort 0). This would likely be an awfully basic Turing machine. Q(1) would speak to the negligible code for a well-ordering with one component. Q( ) would compare to the negligible code for a Turing machine computing the standard requesting on the normal numbers. As gets to be bigger and more complex, requiring more complicated settled structures (e.g., limits, wholes, items of ordinals), the Turing machines computing their comparing well-orderings would moreover have to be be more complex, possibly driving to bigger Gödel numbers. This increment in complexity may be a reflection of the "computational profundity" required to "reach" or "build" the ordinal viably. Computability of Q Is Q itself a computable work? That's , given a recursive ordinal (maybe through an record for a recursive well-ordering of sort), can we successfully compute Q( )? The reply is for the most part no. The pivotal trouble lies in two undecidable issues: Stopping Issue: It is undecidable whether an subjective Turing machine stops on a given input. Typically compounded by the ought to confirm that a given M really "computes a well-ordering". This includes checking properties like transitivity, asymmetry, totality, and most basically, well-foundedness (that each non-empty subset includes

a slightest component). The property of being a well-ordering isn't computable. Arrange Sort Issue: Indeed in case a Turing machine computes a well-ordering, deciding its correct arrange sort is by and large undecidable. There's no calculation that, given a code M for a Turing machine computing a well-ordering (s, s), can yield its arrange sort . Since of these crucial undecidability comes about in computability hypothesis, Q cannot be a computable work in common. Able to characterize Q( ) as a set-theoretic least, but we cannot successfully compute this least for an subjective recursive ordinal . This non-computability underscores the inborn complexity of mapping recursive ordinals to their negligible computational representations. Illustrations and Illustrative scenarios To concretize the thought of Q( ), let us consider a couple of conceptual illustrations. Q(0): The ordinal is the arrange sort of the purge set. A Turing machine M that computes a well-ordering of sort would viably be one that basically chooses that the space s is purge. such a machine would be amazingly straightforward. For any input x, it would report that x / s. The negligible Gödel code for such a basic machine would speak to Q(0). This code would likely be exceptionally little. Q(1): The ordinal 1 is the arrange sort of any singleton set, say 0, with the minor requesting. A Turing machine M 1 for this would choose that s=0 which there are no arrange relations other than 0=0. This is often still a really basic machine, and Q(1) would likely be a little characteristic number, conceivably somewhat bigger than Q(0). Q( ): The ordinal is the arrange sort of the standard requesting on the common numbers (N, work and decides the standard requesting. As we move to more complex recursive ordinals, such as +1, 2, , or , the complexity of the desired Turing machines for the most part increments. For +1, the Turing machine would ought to compute the standard characteristic numbers additionally an extra "final" component. For 2, it would basically concatenate two duplicates of . The development of Q( ) can be seen as a degree of how computationally "costly" it is to develop or recognize a well-ordering of sort .

Associations to Broader Scientific Concepts The definition and suggestions of Q( ) are profoundly interwoven with a few progressed zones of numerical rationale and computability. Viable Clear set Hypothesis Successful graphic set hypothesis, a department of numerical rationale, thinks about determinable sets of normal numbers and reals utilizing the devices of computability hypothesis. Recursive ordinals play a basic part in classifying the complexity of these sets. For occasion, the Wadge progression, a fine-grained chain of command of pointclasses, employments ordinals to degree their complexity. The work Q( ) gives a numerical degree for the complexity of speaking to these recursive ordinals themselves. In this setting, Q( ) can be seen as evaluating the "number-crunching complexity" of inside the successful setting. Understanding the negligible code required to display an ordinal sheds light on how computationally available that ordinal is. For assist perusing on this theme, counsel works by Y. N. Moschovakis, especially his foundational content Expressive set Hypothesis. Later headways in compelling graphic set hypothesis proceed to investigate the complex connections between computational complexity and ordinal progressions; investigating modern investigate on arXiv, such as papers labeled with "successful graphic set hypothesis" or "computable examination," might uncover assist connections. For case, later works on viable adaptations of classical hypotheses, like those concerning Borel sets or explanatory sets, frequently depend on an understanding of recursive ordinals and their computability-theoretic properties. Confirmation Hypothesis and Ordinal Investigation Confirmation hypothesis, especially the subfield of ordinal examination, employments ordinals to degree the consistency quality of formal speculations. Bigger ordinals compare to more grounded speculations. Recursive ordinals, particularly those up to 1 CK , are central to the ordinal examination of subsystems of number-crunching, such as Peano Number-crunching (Dad) and its different parts. The presence of a Turing machine to compute a well-ordering of sort is straightforwardly pertinent to the

idea of a ”documentation framework” for ordinals, which are regularly utilized in proof-theoretic contentions. Whereas Q( ) isn’t a documentation framework in itself (it’s a single common number, not a framework of documentations for all ordinals underneath 1 CK), it speaks to the negligible ’program size’ to recognize . This interfaces to the meta-mathematical think about of formal frameworks: a hypothesis competent of demonstrating the well-foundedness of an requesting of sort certainly requires sufficient ”quality” to bargain with the computational complexity implanted in . The negligible Gödel number Q( ) might be seen as an roundabout degree of the least ”syntactic complexity” required for a formal framework to characterize or analyze an requesting of sort . For significant experiences into ordinal examination, Martin Rathjen’s work on the domain of ordinal investigation gives an amazing outline. Proceeded investigate in verification hypothesis, especially papers on arXiv that dive into ”ordinal investigation” and ”switch arithmetic,” as often as possible use the properties of recursive ordinals to set up foundational comes about approximately the quality of different coherent frameworks. Algorithmic Arbitrariness and Kolmogorov Complexity Whereas not a coordinate application, the soul of negligibility inborn in Q( ) reverberates with concepts from algorithmic haphazardness, especially Kolmogorov complexity. Kolmogorov complexity K(x) of a limited parallel string x is characterized as the length of the most limited program that produces x. This idea measures the characteristic haphazardness or complexity of a string. additionally, Q( ) looks for the negligible Gödel number (which can be seen as a intermediary for program length) for a Turing machine that successfully ”produces” or ”recognizes” the arrange sort . Both concepts point to discover the foremost brief portrayal. For Q( ), the protest being depicted is the arrange structure of a recursive ordinal, instead of a limited string. The association is conceptual: finding the littlest computational outline. For a comprehensive presentation to Kolmogorov complexity, counsel An Presentation to Kolmogorov

Complexity and Its Applications by Ming Li and Paul Vitányi. Later papers on arXiv concerning ”algorithmic data hypothesis” or ”Kolmogorov complexity” regularly investigate expansions and applications of these concepts that might give practically equivalent to systems for understanding the ”data substance” of unbounded numerical objects like recursive ordinals. Viable Ordinal Documentations and Kleene’s O The definition of Q( ) is on a very basic level tied to the viable identification of recursive ordinals and the concept of ordinal documentations. Kleene’s O may be a recursive set of common numbers that serves as a framework of documentations for all recursive ordinals. Each number in O indicates a particular recursive ordinal, and there’s a computable connection that characterizes the ”less than” connection between documentations in O. Q( ) varies from components of O in that it isn’t a ”documentation” within the sense of being portion of a bigger framework that permits for comparison and number-crunching operations. Instep, Q( ) may be a particular numerical esteem related with a recursive ordinal, speaking to the negligible computational exertion to realize that ordinal’s structure. It’s a degree inferred from the presence of such viable representations. The concept of Q( ) highlights the reality that indeed inside the computable domain, there are diverse degrees of ”computational ease” in speaking to ordinals. The Part of Limited Character The choice of the least Gödel number is pivotal and depends on the limited character of Turing machine codes. Each Turing machine can be alloted a special normal number. This permits us to look through them in an climbing arrange: M ,M 1 ,M 2 ,..., where M i =i. For a given recursive ordinal , we are ensured that there exists at slightest one Turing machine M that computes a computable well-ordering of sort . since such machines exist, the set M M computes a computable well-ordering of sort could be a non-empty subset of N. By the well-ordering rule for common numbers, this set must have a interesting slightest component. This slightest component is accurately Q( ). The limited nature of Turing machine depictions, in spite

of their capacity to compute over interminable spaces, is what makes this negligible esteem well-defined. Each Turing machine could be a limited protest, a arrangement of images shaping a program. This finitary portrayal can be encoded as a single characteristic number. The challenge, as famous, isn't within the definition of Q( ) but in its potential computability or the capacity to successfully discover this negligible machine in hone. Numerical Profundity and Open Questions The work Q gives a focal point through which to look at the inborn computational substance of recursive ordinals. It moves past only declaring that an ordinal is recursive to inquiring how effectively it can be seen as such. This opens up a few regions for more profound examination: Fine-grained Complexity of Ordinals: Can Q be utilized to set up a more fine-grained pecking order of recursive ordinals than conventional approaches? Whereas the ordinals themselves frame a well-ordered chain of command, their computational representations might uncover distinctive viewpoints of complexity. Relationship to standard Progressions: How does Q( ) relate to particular chains of command in computability hypothesis, such as the

arithmetical chain of command, the explanatory pecking order, or the hyperarithmetical chain of command? The components of Re are profoundly associated to the structures inside these pecking orders. Computational Analogs: May a comparable work be characterized for other unbounded computable structures, such as recursive trees, recursive charts, or recursive straight orderings? The system of finding a negligible Gödel number for a computing operator appears broadly pertinent. Guess of Q( ): since Q is by and large uncomputable, can we discover compelling upper bounds or approximations for Q( ) for certain classes of recursive ordinals? For occasion, for

Chunck 3

Anasjellta e majtë

one can define a map

$$\mathcal{W} : \text{range}(\mathcal{Q}) \to \text{Re}$$

is a function such that

$$\mathcal{W} \circ \mathcal{Q} = \text{Id}_{\text{range}(\mathcal{Q})}$$

The space of recursive ordinals, regularly signified as Re or identically as the components of 1 CK (the Church-Kleene ordinal), speaks to the collection of all countable ordinals that can be realized as the arrange sort of a computable well-ordering on a subset of the normal numbers. These ordinals are crucial in computability hypothesis, serving as a degree of complexity for different viable forms and pecking orders. A pivotal angle of their consider includes evaluating their computational ”fetched” or ”address,” driving to the definition of particular mappings from ordinals to normal numbers and vice-versa. Central to this request are the capacities Q and its conceptual converse, W, which together set up a canonical encoding framework for recursive ordinals. The work Q:Re→N maps each recursive ordinal to a one of a kind characteristic number. This mapping is characterized

with a solid accentuation on negligibility in computational representation: [beginalign* mathcalQ(alpha) = minleftulcorner mathcalMurcorner mid mathcalM content computes a computable well-ordering prec$_s$$contentonssubseteqmathbbNtextsuchthatright.clearedoutexistsf : stoalphacontentsuchthat(fcontentmaybeabijectionarriveforallx, yins(xprec_syif$
$f(y)))rightendalign*]Toexpound, awell-orderingsonasubsetsNisconsideredcom$
$orderingintheeventthatitsprogram, formalizedbyaGdelnumberMN, successfully$
$isomorphism, guaranteeingthatthewell-orderedstructure(s, s)hasanarrangesor$
$orderingandcomparingTuringmachineMareensuredtoexist.Thecharacterizingin$
$orderingisomorphicto.Thepresenceofthisleastisguaranteedbythewell-$
$orderingguidelineofthecommonnumbers.ThisnegligibleGdelnumbersuccessfully$

the ordinal can be computationally realized employing a generally less complex calculation. For occurrence, Q(0) would compare to the negligible Gödel number for a Turing machine computing the purge well-ordering, likely an awfully little number. Alternately, Q( ), the negligible code for a machine computing the standard requesting on N, would be essentially bigger, reflecting the expanded algorithmic complexity. A pivotal property determined from this definition is the injectivity of Q. On the off chance that Q( )=Q( ) for , Re, let k=Q( )=Q( ). This implies k is the Gödel number of a interesting Turing machine M k . since any single Turing machine computes a particular, special well-ordering (on the off chance that it computes one at all), it must compute a well-ordering of a interesting arrange sort. Hence, in case M k computes a well-ordering of sort , it cannot at the same time compute a well-ordering of a unmistakable sort . Hence, must break even with , demonstrating Q is injective. In spite of its exact definition and injectivity, the work Q is for the most part uncomputable. This emerges from crucial undecidability comes about in computability hypothesis. To compute Q( ) for an self-assertive , one would ought to: efficiently list Turing machines by their Gödel numbers. For each machine M i , decide in case it computes a computable well-ordering. This includes undecidable issues just like the Halting Problem, as checking

well-foundedness isn't for the most part algorithmic. In the event that M i computes a well-ordering, decide its arrange sort. This, as well, is an undecidable issue; there's no common calculation to decide the arrange sort of an self-assertive recursive well-ordering. These computational obstructions cruel that whereas Q( ) is scientifically well-defined for each recursive ordinal, there's no compelling strategy to discover its esteem. Complementing Q is the work W:range(Q)→Re, which is characterized by the property: [mathcalW circ mathcalQ = mathrmId$_{mathrmrange(mathcalQ)}$]$Thisimpliesthatforanyrecursiveordinal, intheeventthatk$
$Q(), atthatpointW(k) = .Inquintessence, Wtakesacommonnumberkfromtheextraordin$
$definedonitsspace : eachkrange(Q)comparestopreciselyonerecursiveordinal.LikeQ, the$
$orderingitcomputes.Asbuiltup, decidingthearrangesortofanself-assertivecomputablew$
$orderingisundecidable.Inthismanner, nocalculationcancomputeW(k)incommon.Thecon$
$whereasrecursiveordinalscanbeviablyspokento, themethodoffindingtheirnegligiblerep$
$theoreticordinalsandtransfiniteacceptancestandards).Notatalllikecustomaryordinald$

Chunck 4

sekuenca themelore

one can define a map

$$\begin{aligned} \text{maxel} &: \mathcal{P}(\text{Re}) \to \text{Re} \\ \alpha &\mapsto \text{maxel}(\alpha) \\ \text{maxel}(\alpha) &= \beta \text{ such that } \beta \in \alpha \wedge \forall x \in \alpha (x \leq \beta) \end{aligned}$$

one can define a map

$$\begin{aligned} \mathcal{W}^{-1} &: \text{Re} \to \text{range}(\mathcal{Q}) \\ \mathcal{W}^{-1} \text{ is a function such that} & \\ \mathcal{W}^{-1} \circ \mathcal{W} = \text{Id}_{\text{Re}} & \\ \wedge \mathcal{W} \circ \mathcal{W}^{-1} = \text{Id}_{\text{range}(\mathcal{Q})} & \end{aligned}$$

one can define a map

$$\begin{aligned} \mathcal{E} &: \text{range}(\mathcal{Q}) \times \mathbb{N} \to \mathbb{N} \\ (i, j) &\mapsto \mathcal{E}(i, j) \end{aligned}$$

$$\mathcal{E}(i,j) = \begin{cases} i & \text{if } \mathcal{W}(i) = 0 \\ \mathcal{W}^{-1}(\text{maxel}(\mathcal{W}(i))) & \text{if } \exists \alpha : \mathcal{W}(i) = \alpha + 1 \\ \mathcal{W}^{-1}(\{0\} \cup \text{maxel}(\{\gamma \in \mathcal{W}(i) \mid \mathcal{W}^{-1}(\gamma) < j\}) + j) & \text{if } \neg\exists \alpha : \mathcal{W}(i) = \alpha + 1 \end{cases}$$

Typical Investigation of a Framework for Recursive Ordinal Control I. Presentation to the Formal Framework This report presents a comprehensive formal examination of a set of numerical definitions concerning recursive ordinals and operations upon them. The definitions present a few mappings: Q, W, maxel, W -1 , and E, which show up to develop a system for dealing with recursive ordinals and their representations inside the normal numbers. The overarching objective is to fastidiously clarify each definition, dismember its components, and thoroughly recognize any inborn botches, circular rationale, typographical blunders, ill-defined terms, or other consistent irregularities. This basic examination is grounded in built up standards of numerical rationale, computability hypothesis, and set hypothesis, guaranteeing a tall degree of formal accuracy. A strong understanding of computability hypothesis, counting Turing machines and Gödel numbering, as well as the hypothesis of ordinals, especially recursive ordinals and well-orderings, is prerequisite for a full appreciation of this investigation. These foundational concepts are expounded upon within the subsequent segment to set up a common specialized ground for the thorough assessment of the proposed framework. II. Foundational Concepts in Computability Hypothesis and Ordinal Number juggling The formal framework beneath examination works on crucial concepts from computability hypothesis and ordinal math. A exact understanding of these foundational components is fundamental for a exhaustive assessment of the system's inner consistency and numerical soundness. Recursive Ordinals (Re) The lesson Re signifies the collection of all recursive ordinals. A recursive ordinal is formally characterized as the arrange sort of a few computable well-ordering of a computable subset of the common numbers. This definition sets up a essential interface between ordinal hypothesis and computability hypothesis, attesting that such ordinals are those whose structure can be successfully spoken to by an calculation. The set of all recursive ordinals shows a pivotal property: it is closed downwards. This implies that in the event

that an ordinal is recursive, all ordinals entirely littler than it are too recursive. This property is vital for the well-foundedness of recursive definitions over ordinals, guaranteeing that transfinite developments end fittingly. The supremum of all recursive ordinals may be a critical countable ordinal known as the Church-Kleene ordinal, indicated 1 CK . This ordinal speaks to the primary ordinal that's not recursive, meaning that it cannot be the arrange sort of any computable well-ordering of the common numbers. The Church-Kleene ordinal serves as a basic boundary within the chain of command of countable ordinals, stamping the constrain of what can be viably depicted

or computed inside the system of Turing machines. The definition of Re as "the lesson of all recursive ordinals" verifiably sets up that Re is absolutely the set of all ordinals entirely less than 1 CK . Usually not just a graphic property but a definitional boundary for the course Re. The course Re is hence a particular beginning fragment of the ordinals. This exact boundary is vital since it limits the scope of the capacities characterized within the issue. Any work whose space is Re is verifiably limited to ordinals underneath 1 CK . Ought to any operation inside the framework create an ordinal break even with to or more prominent than 1 CK , it would drop exterior the characterized space of Re, driving to indistinct behavior or a breakdown of the system's inner consistency. This builds up a difficult restrain on the "computational control" of the ordinal operations inside the framework. Assist discourses on the computability on the space of countable ordinals and recursively built ordinals strengthen the idea that recursive ordinals are those agreeable to algorithmic control. Computable Well-Orderings A well-ordering s on a subset s N is regarded computable in the event that the connection x s y (and its complement, x s y) could be a computable connection on N×N. This requires the presence of a Turing machine that, given any two components x,y s, can choose in a limited number of steps whether x s y. Besides, the set s itself must be computable (recursive). The arrange sort of such a

well-ordering is the interesting ordinal number isomorphic to it. For a connection to be "computable," it ordinarily suggests that the characteristic work of the connection is add up to computable. This implies the Turing machine computing the connection must end on all inputs from its space (in this case, sets from s×s). On the off chance that the machine as it were in part computes the connection (i.e., it might not stop for a few sets), the well-ordering would not be considered "computable" within the solid sense required for recursive ordinals. This add up to computability necessity forces a solid limitation. It guarantees that the arrange connection s is completely decidable. Any uncertainty or non-halting condition for the Turing machine M within the definition of Q would render the well-ordering non-computable, and in this way the ordinal not recursive, in a general sense undermining the complete framework. This highlights the foundational dependence on the Church-Turing

proposal, which sets the proportionality of successfully calculable capacities with those computable by a Turing machine. Different inquire about endeavors examine "well-ordering standards" and their association to computability hypothesis, especially in Invert Science, underscoring the profound transaction between well-orderings and computability. Gödel Numbering of Turing Machines Gödel numbering gives a precise and viable (computable) way to relegate a special common number, frequently alluded to as a Gödel number or list, to each Turing machine. This handle includes encoding the machine's limited depiction, which incorporates its letter set, states, and move rules, into a single characteristic number. This encoding is regularly one-to-one, guaranteeing that particular Turing machines have unmistakable Gödel numbers. The documentation M speaks to the Gödel number of a particular Turing machine M. This numerical representation encourages the control and ponder of Turing machines inside math, shaping the exceptionally premise of computability hypothesis and metamathematics. The presence of such an compelling identification may

be a foundation for characterizing capacities that work on programs or their properties, as seen within the common documentation e for the e-th fractional computable work, which verifiably depends on a Gödel numbering conspire. Whereas different Turing machines can compute the same work or connection, and numerous computable well-orderings can have the same arrange sort, the definition of Q unequivocally looks for the least Gödel number M . This "least" operation viably chooses a canonical agent for each recursive ordinal inside the space of Turing machine codes. This suggests that for any given recursive ordinal , there's a special littlest Gödel number that compares to a Turing machine computing a well-ordering of sort . This canonical choice is fundamental for Q to be a well-defined work, guaranteeing it yields a special yield for each input. Without this "least" clause, Q would be a multi-valued connection, not a work. This canonicalization may be a common procedure in computability hypothesis to set up interesting representations for scientific objects that might something else have numerous identical computational depictions. It moreover suggests that the set of Gödel numbers speaking to recursive ordinals (the run of Q) shapes a particular, well-defined subset of N. III. Point by point Examination of Characterized Mappings The formal framework beneath survey comprises a few interconnected mappings, each planned to function on recursive ordinals or their numerical representations. A fastidious examination of each mapping is vital to find out the system's by and large coherence and numerical meticulousness. A. Examination of Q:Re→N The work Q is formally characterized as:

This definition indicates that Q maps a recursive ordinal to the littlest Gödel number of a Turing machine M. This machine M must compute a computable well-ordering s on a few computable subset s of common numbers. The basic condition is that this well-ordering s must have the arrange sort . Usually guaranteed by the presence of an order-preserving bijection f:s→ . The incorporation of the "least" administrator may be a essential angle of this definition, because it

guarantees that Q( ) yields a interesting characteristic number for each Re. This uniqueness is irreplaceable for Q to be a well-defined work. By the exceptionally definition of a recursive ordinal, for each Re, there exists at slightest one computable well-ordering on a subset of N with arrange sort . Since Turing machines can be successfully counted , there will be a non-empty set of Gödel numbers comparing to machines that compute such well-orderings. As this set of Gödel numbers constitutes a non-empty subset of N, it must have a least component by the well-ordering rule of normal numbers. Thus, Q( ) is continuously well-defined for any Re. This work viably allocates a canonical Gödel number to each recursive ordinal. This can be a standard and capable approach in computability hypothesis to "arithmetize" theoretical numerical objects like ordinals, subsequently permitting them to be controlled by computable capacities. This prepare is closely resembling to the development of Kleene's O framework of ordinal documentations, which gives a canonical system for speaking to recursive ordinals. The work Q maps recursive ordinals to common numbers, and by creating a least Gödel number for each ordinal, it viably makes a canonical "documentation" for each recursive ordinal. This is often not merely an subjective mapping, but one that's inherently connected to the computational complexity (as reflected by the Gödel number) required to speak to the ordinal. On the off chance that Q is undoubtedly a canonical documentation framework, it infers that Q must be injective. In the event that Q(1)=Q(2), at that point 1 and 2 must be the same ordinal, since the least Gödel number interestingly recognizes the arrange sort. This injectivity will demonstrate pivotal for the properties of ensuing capacities, especially W and W −1 . Moreover, this suggests that the extend of Q may be a particular, computably enumerable (in spite of the fact that not essentially computable) set of common numbers. The properties of Q are summarized in Table III.A.1. Property Portrayal Space Re (Course of all recursive ordinals) Codomain N (Characteristic numbers) Input Sort Recursive

ordinal Yield Sort Characteristic number (Gödel number) Key Property Maps each recursive ordinal to the least Gödel number of a Turing machine computing a well-ordering of that sort. Well-definedness Status Well-defined for all Re due to the well-ordering of N and the definition of recursive ordinals. Trade to Sheets Table III.A.1: Properties of Q B. Investigation of W:range(Q)→Re The work W is characterized as: and is indicated to be a work such that [mathcalW circ mathcalQ = mathrmIdmathrmrange(mathcalQ)] This definition states that W maps a characteristic number i (which must have a place to the extend of Q, meaning i may be a canonical Gödel number of a few recursive ordinal) back to the recursive ordinal it speaks to. The condition $mathcalWcircmathcalQ = mathrmIdmathrmrange(mathcalQ)$ suggests that for any Re, applying Q to get its Gödel number, and hence applying W to that Gödel number, reestablishes the initial ordinal . Particularly, for any Re, W(Q( ))= . The expressed property straightforwardly characterizes W as a cleared out reverse of Q. For a cleared out reverse to exist and be a well-defined work, the initial work (Q) must be injective. As already built up, Q is in fact injective due to its "least" clause, which guarantees that each recursive ordinal maps to a one of a kind Gödel number. This injectivity ensures that each Gödel number in range(Q) compares to precisely one recursive ordinal, subsequently rendering W interestingly decided. The property W(Q( ))= suggests that W viably "interprets" the canonical Gödel number back into the ordinal it speaks to. This absolutely mirrors the part of an order-type work. On the off chance that i=Q( ), at that point i is the Gödel number of a Turing machine that computes a well-ordering of sort . Subsequently, W(i) must be . This affirms W as the work that takes a Gödel number of a computable well-ordering and returns its comparing arrange sort. The presence and well-definedness of W imply that the mapping from canonical Gödel numbers to recursive ordinals is unambiguous, a essential property for any successful framework of ordinal documentations. This suggests that the structure of recursive

ordinals is loyally captured by their canonical Gödel numbers. The properties of W are summarized in Table III.B.1.

| Property | Description | | :— | :— | | Domain | range(Q) (Set of canonical Gödel numbers of recursive ordinals) | | Codomain | Re (Class of all recursive ordinals) | | Input Type | Natural number (canonical Gödel number) | | Output Type | Recursive ordinal | | Key Property | Left inverse of Q; maps a canonical Gödel number to the recursive ordinal it represents. | | Well-definedness Status | Well-defined and unique, contingent on the injectivity of Q. | Table III.B.1: Properties of W C. Examination of maxel:P(Re)→Re The work maxel is formally characterized as: This work is aiming to distinguish the greatest component inside a given set of recursive ordinals. The required space, P(Re), indicates the control set of all recursive ordinals, meaning that within the definition speaks to a set of recursive ordinals, not a single ordinal. The component is characterized as the greatest in the event that it could be a part of the set and more noteworthy than or break even with to all other components x inside . A basic examination uncovers that this definition is in a general sense ill-defined for any set P(Re) that does not have a most extreme component. Numerous sets of ordinals, especially those speaking to restrain ordinals or unbounded groupings, intrinsically need a greatest component. For occurrence, on the off chance that is the set of all limited ordinals, i.e., =0,1,2,...= (when seen as a set of ordinals), there's no most extreme component in . Additionally, on the off chance that = , +1, +2,..., this set too contains no greatest. Any unbounded expanding arrangement of ordinals inside Re, such as the grouping (n) n N which meets to , would shape a set without a greatest component. In such cases, no fulfilling the definition can be found, rendering maxel( ) vague. A encourage point of concern lies within the variable naming. The utilize of as a variable for the input set in maxel( ) is possibly befuddling, as is customarily utilized to represent a single ordinal. Whereas linguistically reasonable, this choice makes a potential for error, particularly when considering

its ensuing utilization within the definition of E, where W(i) (which is an ordinal) is passed as an contention to maxel. On the off chance that, opposite to the space determination P(Re), were in fact planning to be a single ordinal (as proposed by its application in E), the definition would still be tricky. In the event that $=+1$ (a successor ordinal), its components are has no biggest component among its forerunners. Thus, there's no such that x (x). In this elucidation, the work would be unclear for all restrain ordinals. The work maxel endures from extreme ill-definedness. It cannot work on self-assertive subsets of Re (its announced space) since numerous such subsets need a most extreme component. Besides, in case it were aiming to function on single ordinals (as its documentation in E recommends), it would stay ill-defined for restrain ordinals. This speaks to a crucial misguided judgment of ordinal structure. A center property of restrain ordinals is that they are the supremum of their forerunners, and hence the "set of components" that constitute the ordinal itself does not contain a most extreme. The definition of maxel specifically negates this principal property for restrain ordinals. This proposes a misconception of how ordinals, especially restrain ordinals, are organized. Typically not a insignificant typographical blunder but a profound conceptual blemish with respect to ordinal number juggling. Any work depending on maxel for restrain ordinals will be unclear or create erroneous comes about. This specifically impacts the definition of E, which is expecting to characterize crucial arrangements, a concept intrinsically tied to restrain ordinals. The framework, as displayed, cannot accurately handle restrain ordinals due to this imperfection. The examination of maxel is summarized in Table III.C.1. Property Depiction Issue Space P(Re) (Control set of recursive ordinals) Ill-defined for sets without a most extreme component. Codomain Re (Lesson of all recursive ordinals) Yield is unclear in case no greatest exists. Input Sort Set of recursive ordinals Perplexity with single ordinal input in ensuing employments. Yield Sort Recursive ordinal Vague for numerous substantial inputs.

Key Property Planning to discover the greatest component of a set. Comes up short for boundless expanding arrangements and restrain ordinals (seen as sets of their forerunners). Well-definedness Status Ill-defined for a critical parcel of its space. Case: maxel(0,1,2,...) is unclear. Send out to Sheets Table III.C.1: Examination of maxel Work D. Investigation of W −1 :Re→range(Q) The work W −1 is formally characterized as: This definition expressly states that W −1 is the two-sided converse of W. For W −1 to exist as a well-defined work with these properties, W must be a bijection from its space (range(Q)) to its codomain (Re). This requires W to be both injective and surjective. A consistency check with past definitions affirms the bijective nature of W. To begin with, consider the injectivity of Q. As built up within the investigation of Q,this work is injective since it maps each recursive ordinal to a interesting least Gödel number. Moment, consider the surjectivity of W. Since Q:Re→range(Q) is surjective by the exceptionally definition of range(Q), and the condition W Q=Id range(Q) suggests W(Q( ))= for all Re, it takes after specifically that W is surjective onto Re. For any Re, there exists an i=Q( ) range(Q) such that W(i)= . Third, consider the injectivity of W. Assume W(i 1)=W(i 2) for a few i 1 ,i 2 range(Q). Since i 1 ,i 2 range(Q), there must exist 1 , 2 Re such that i 1 =Q(1) and i 2 =Q(2). At that point, the uniformity W(Q(1))=W(Q(2)) infers 1 = 2 (by the cleared out converse property). Since Q is injective, Q(1)=Q(2) infers i 1 =i 2 . Subsequently, W is injective. Given that W is both injective and surjective, it is undoubtedly a bijection from range(Q) to Re. Thus, its two-sided reverse W −1 is well-defined and exists absolutely as expressed within the formal definition. The effective definition of W −1 as a two-sided reverse affirms that W may be a bijection. This, in turn, infers that Q sets up a one-to-one correspondence between the set of all recursive ordinals (Re) and the set of their canonical Gödel numbers (range(Q)). This is often a solid result, illustrating that recursive ordinals can be extraordinarily and unam-

biguously encoded and decoded as common numbers. This bijective relationship is crucial to the exceptionally idea of "computable" or "recursive" ordinals. It implies that the whole structure of Re is viably reflected inside a particular subset of N, permitting for algorithmic control and consider of these transfinite objects utilizing limited computational implies. This is often a foundation of compelling arithmetic. The properties of W −1 are summarized in

Table III.D.1. | Property | Description | | :— | :— | | Domain | Re (Class of all recursive ordinals) | | Codomain | range(Q) (Set of canonical Gödel numbers of recursive ordinals) | | Input Type | Recursive ordinal | | Output Type | Natural number (canonical Gödel number) | | Key Properties | Two-sided inverse of W; establishes a bijection between recursive ordinals and their canonical Gödel numbers. | | Well-definedness Status | Well-defined and

unique, given the bijective nature of W. | Table III.D.1: Properties of W −1 E. Analysis of E:range(Q)×N→N The function E is defined as: [

$$\mathcal{E} : \text{range}(\mathcal{Q}) \times \mathbb{N} \to \mathbb{N} \ (i, j) \qquad\qquad \mapsto \mathcal{E}(i, j)$$

] [E(i,j) = $\begin{cases} i & \text{if } \mathcal{W}(i) \\ \text{if } \exists\alpha : \mathcal{W}(i) = \alpha + 1 \ \mathcal{W}^{-1}(0 \cup \text{maxel}(\gamma \in \mathcal{W}(i) \mid \mathcal{W}^{-1}(\gamma) < j) + j) & \text{if } eg\exists\alpha \end{cases}$

] This work takes a Gödel number i (speaking to an ordinal) and a characteristic number j, and returns a normal number. It shows up to be an endeavor to characterize a framework of crucial arrangements or a related ordinal math operation on Gödel numbers, which are significant for transfinite recursion. A case-by-case examination uncovers a few noteworthy issues: Case 1: W(i)=0 Definition: E(i,j)=i. Investigation: In case the ordinal spoken to by i is 0, at that point E essentially returns i. Typically reliable with the mapping Q(0) being the canonical Gödel number for the ordinal 0. This case accurately handles the base ordinal in a framework of essential arrangements. Case 2: :W(i)= +1 Defi-

nition: E(i,j)=W −1 (maxel(W(i))). Examination: In case W(i) could be a successor ordinal, say +1, at that point maxel(W(i)) (accepting maxel is translated to apply to an ordinal and accurately yields its forerunner, as examined in Area III.C) would result in . Hence, E(i,j)=W −1 ( ). This operation maps the Gödel number of a successor ordinal to the Gödel number of its forerunner. Typically a standard component of crucial grouping definitions for successor ordinals, where +1 is approximated by . This case shows up to be expecting to compute the Gödel number of the forerunner ordinal. Case 3: ”eg :W(i)= +1” Definition: E(i,j)=W −1 (0 maxel(W(i) W −1 ( ) that this case was planning for constrain ordinals. In standard definitions of principal groupings, a constrain ordinal is approximated by a arrangement [j] for j N, frequently including a supremum over forerunners. Point by point Assessment of the Expression (accepting W(i) could be a restrain ordinal , as likely planning): Sub-expression: W(i) W −1 ( ) a most extreme component, causing maxel to come up short, and thus, E to be indistinct for these significant inputs. This chain of reliance implies that the whole framework, especially its capacity to handle constrain ordinals, is on a very basic level broken. A strong framework of principal arrangements is fundamental for transfinite recursion and characterizing operations on ordinals past successors. Without a legitimately characterized component for constrain ordinals, the system's utility for progressed ordinal number-crunching is extremely compromised. The typo within the case condition could be a shallow mistake, but the fundamental conceptual blemish in maxel is disastrous for E. The structure of the third case of E, particularly ”W −1 (0 maxel(W(i) W −1 ( ) Issues Typographical/Logical Mistake in E's Case Conditions: The foremost clear issue is the indistinguishable condition for the moment and third cases of the E work. The condition ”eg :W(i)= +1” could be a verbatim reiteration of the condition for the successor ordinal case. This coherent excess guarantees that the third case, which is basically planned to handle constrain ordinals, is never come to. This

avoids the framework from legitimately characterizing crucial groupings for a vital course of ordinals. Ill-defined Work maxel: The work maxel:P(Re)→Re is characterized to discover the most extreme component of a set of recursive ordinals. In any case, this definition is intrinsically tricky since numerous sets inside its indicated space don't have a greatest component. For occurrence, any boundless expanding grouping of recursive ordinals, or the set of all forerunners of a constrain ordinal (which is the restrain ordinal itself when seen as a set of its components), on a very basic level needs a most noteworthy component. Subsequently, maxel is vague for a noteworthy parcel of its space, driving to a halfway work where a add up to work is suggested by its codomain. Proliferation of maxel's Ill-definedness to E: The work E fundamentally depends on the maxel work in its third case. Since maxel is ill-defined for sets speaking to restrain ordinals (which this case is assumed to handle), E gets to be unclear for inputs i where W(i) could be a restrain ordinal. This coordinate reliance implies that the conceptual imperfection in maxel cascades, rendering E non-total and scientifically unsound for a principal lesson of inputs. A framework expecting to function over all recursive ordinals must heartily handle restrain ordinals, which this framework right now comes up short to do. Uncertainty in maxel's Input Sort: Whereas formally pronounced with a space of P(Re) (control set of recursive ordinals), the variable is utilized as the input variable for maxel( ). Its ensuing application inside E as maxel(W(i)), where W(i) may be a single ordinal, recommends an underlying confusion with respect to whether maxel is aiming to function on sets of ordinals or on person ordinals (translated as the set of their forerunners). Indeed on the off chance that deciphered as working on an ordinal, the work remains ill-defined for restrain ordinals, as examined. Uncertain Documentation in E's Third Case: The expression "0 maxel(...)+j" within the third case of E utilizes documentation that's not standard for ordinal number-crunching. In case maxel were to return an ordinal , at that point 0 isn't a routine

ordinal operation. It is likely planning to speak to max(0,) or

to guarantee the set for a consequent supremum operation is non-empty. Such notational imprecision can lead to error and uncertainty in formal settings. Suggestions for Redress and Clarification To correct the distinguished issues and set up a numerically sound and compelling framework for recursive ordinal control, the taking after adjustments and clarifications are basic: For E's Case Conditions: The third case condition must be rectified to consistently recognize it from the successor case. The foremost characteristic and scientifically suitable adjustment, given the structure of the expression, is to characterize it for restrain ordinals: [textif mathcalW(i) content may be a restrain ordinal] This guarantees that all sorts of recursive ordinals (zero, successor, constrain) are dealt with by particular, non-overlapping cases. For maxel and its utilize in E: The work maxel must be in a general sense re-imagined or totally supplanted. Choice 1 (Rethink maxel): In the event that the aim is entirely to discover the biggest component of a limited set of ordinals, the space of maxel ought to be unequivocally limited to limited subsets of Re. In any case, this would not serve the reason of building essential arrangements for constrain ordinals, which inalienably include possibly boundless sets of forerunners. Alternative 2 (Supplant with Supremum): For dealing with constrain ordinals in principal groupings, the maxel operation ought to be supplanted by a supremum (slightest upper bound) administrator. The supremum of any non-empty set of ordinals continuously exists and is well-defined. The expression in E's third case would at that point gotten to be: [mathcalW$^{-1}(sup(0glassgammainmathcalW(i)midmathcalW^{-1}(gamma) < j)+ j)]Thisrequiresawell-definedcomputablesupremumworkforsetsofrecursiveord$ $definednessissue.ForUncertaintyinmaxel'sInputSort : Clarificationisrequired$ $theoreticdefinitionofordinals, andthedefinitionoughttoexpresslystateitsoperati$ $Intheeventthatthesupremumisutilized, thedocumentation0oughttobemademoref$ $zerocomponents)tododgeequivocalness.Therecognizedissuesandproposedresoluti$

of Issue Point by point Clarification of Issue Proposed Resolu-

tion/Correction E (Case 3 condition) Typo, Consistent Irregularity Condition "eg :W(i)= +1" is indistinguishable to Case 2, making Case 3 inaccessible. Alter condition to "on the off chance that W(i) may be a constrain ordinal". maxel (Definition) Ill-defined Work Characterized to discover greatest of a set, but numerous sets of ordinals (e.g., restrain ordinals' forerunners) have no greatest. Rethink to function as it were on limited sets, or, ideally, supplant with a supremum administrator for common ordinal sets. E (Case 3 expression) Proliferation of Ill-definedness Depends on maxel for restrain ordinals, where maxel is indistinct, making E indistinct for these vital inputs. Supplant maxel with a computable supremum work for sets of recursive ordinals. maxel (Input variable) Equivocalness in Input Sort Variable utilized for both a set of ordinals (in definition) and a single ordinal (in E's application). Clarify planning input sort; in the event that single ordinal, rename variable and rethink for successor/limit cases. E (Case 3 expression) Uncertain Documentation Expression "0 maxel(…)+j" contains non-standard ordinal set union. Reword utilizing standard ordinal operations, e.g., max(0,sup(…))+j. Send out to Sheets Table IV.1: Rundown of Distinguished Issues and Proposed Resolutions V. Conclusion The thorough examination of the displayed formal framework for controlling recursive ordinals uncovers a blended scene of foundational quality and basic definitional shortcomings. The beginning definitions of Q, W, and W -1 effectively set up a steady and bijective mapping between recursive ordinals and their canonical Gödel numbers. This one-to-one correspondence could be a crucial accomplishment, illustrating that the complex structure of recursive ordinals can be steadfastly spoken to and successfully controlled inside the space of common numbers. This bijective relationship could be a foundation of viable science, empowering the arithmetization of transfinite objects. Be that as it may, the framework endures from significant blemishes in its taking care of of ordinal maximums and, subsequently, its capacity to characterize operations for restrain or-

dinals. The work maxel is in a general sense ill-defined for a critical parcel of its pronounced space, especially for sets of ordinals that need a greatest component, such as the forerunners of constrain ordinals. This ill-definedness isn't an separated mistake; it engenders specifically to E, the work aiming to characterize crucial groupings. As a result, E gets to be non-total and numerically unsound for any input comparing to a restrain ordinal. Besides, a clear typographical mistake in E's case qualification worsens these formal irregularities, rendering the expecting restrain ordinal case inaccessible. The recognized issues extremely compromise the hypothetical soundness and commonsense ease of use of the displayed

framework, especially for operations including restrain ordinals. A strong framework for ordinal number juggling and the development of crucial groupings requires exact and all around pertinent definitions for all ordinal sorts (zero, successor, and constrain). As right now defined, the framework cannot dependably compute principal arrangements for constrain ordinals, which could be a center prerequisite for numerous progressed applications in confirmation hypothesis and computability hypothesis, such as transfinite recursion and the calibration of coherent hypotheses. To realize a scientifically sound and successful framework, a total re-evaluation and redefinition of the maxel work (or its substitution by a well-defined supremum administrator) and the case refinements inside E are basic. Tending to these essential issues would change the displayed system from a conceptually promising but formally imperfect develop into a coherent and effective apparatus for the thorough think about of recursive ordinals and their computational properties.

Chunck 5

Virtyti më i lartë nuk është virtyt, kështu që ka virtyt; virtyti më i ulët nuk është pa virtyt, kështu që nuk ka virtyt

One can define a map

$$\begin{aligned} \mathcal{R} : \mathrm{range}(\mathcal{Q}) \times \mathbb{N} &\to \mathbb{N} \\ (i, j) &\mapsto \mathcal{R}(i, j) \end{aligned}$$

$$\mathcal{R}(i, j) = \begin{cases} i & \text{if } \mathcal{W}(i) = 0 \\ \sum_{k=0}^{j} \mathcal{R}^k(\mathcal{E}(i, k), k) & \text{if } \mathcal{W}(i) \neq 0 \end{cases}$$

$$\begin{aligned} \mathcal{T} : \mathbb{N} &\to \mathbb{N} \\ i &\mapsto \mathcal{T}(i) \end{aligned}$$

$$\mathcal{T}(i) = \max\{\mathcal{R}(k, i) \mid k \in \mathbb{N} \ \wedge \ k \leq i\}$$

.

Virtyti më i lartë nuk është asgjë, asgjë nuk bëhet. Mirëdashësia më e lartë bëhet dhe drejtësia më e lartë bëhet e bëhet. Etiketa më e lartë bëhet me krahë e krahë. Prandaj, rruga humbet. Pasi humbet rruga, virtyti humbet dhe pastaj mirëdashja humbet. Pasi humbet mirëdashja, atëherë drejtësia humbet. Fillimi i kaosit është lulja e rrugës dhe fillimi i budallallëkut. Një njeri i madh jeton në Qi të trashë por jo në Qi të qetë, jeton në Qi të fortë por jo në lule Qi, kështu që qëroni lëkurën për ta marrë këtë

Virtyti më i lartë nuk është virtyt, kështu që ekziston virtyti. Virtyti më i ulët nuk e humbet virtytin, kështu që nuk ka virtyt. Virtyti më i lartë nuk është asgjë dhe asgjë nuk bëhet. Mirëdashja më e lartë bëhet por asgjë. Të mendosh është më e mira. Të bësh dhe të kesh një mendim është më e mira. Të bësh dhe askush të mos përgjigjet është të ngresh krahët dhe ta bësh. Prandaj, nëse e humb rrugën, do të kesh virtyt. Nëse e humbet virtytin, do të kesh mirëdashje. Nëse e humbet mirëdashjen, do të kesh drejtësi. Nëse e humb drejtësinë, do të kesh korrektësi. Korrektësia është spiranca e besnikërisë dhe besimit, dhe fillimi i kaosit. Para-njohuria është lulja e rrugës, dhe fillimi i budallallëkut. Prandaj, një njeri i vërtetë jeton në paqe dhe nuk jeton në shkëlqim. Prandaj, ai largohet dhe e merr këtë.

Në të kaluarën, ata që fitonin unitetin, qielli fitonte unitetin për të qenë i kthjellët, toka fitonte unitetin për të qenë i qetë, shpirti fitonte unitetin për t'u larë në shi, dhe unitetin për të qenë plot me markezin Dhe mendonin se është Qi-ja e duhur për ta sjellë atë. Qielli nuk është shumë i kthjellët, dhe do të ketë frikë. Toka nuk ka shumë frikë. Shpirti nuk është shumë me shi, dhe do të ketë frikë të pushojë. Banja nuk është shumë e mbushur, dhe do të ketë frikë të ketë etje. Princat dhe mbretërit nuk janë shumë fisnikë, kështu që ata duhet të jenë fisnikë dhe të marrin të lirën si bazë. Ata duhet të jenë të lartë dhe të marrin të ulëtën si themel. Prandaj, princat dhe mbretërit janë të vetmuar dhe të varfër. Ky Qi është i lirë dhe i gabuar, kështu që çon në numër dhe hiç. Prandaj, nuk është e dëshirueshme të jesh si një gur nefriti.

Në të kaluarën, ata që merrnin një, qielli merrte një për ta pastruar, toka merrte një për ta qetësuar, shpirti merrte një për të rënë shi dhe banja merrte një për t'u mbushur. Princat dhe mbretërit merrnin një dhe mendonin se bota ishte Qi e drejtë. Qielli nuk është shumë i kthjellët, do të ketë frikë nga lotusi. Toka nuk është shumë e qetë, do të ketë frikë nga perënditë. Lugina nuk është shumë e thatë, do të ketë etje. Princat dhe mbretërit nuk janë shumë fisnikë, ata do të kenë frikë të bien. Prandaj, ata duhet të jenë fisnikë dhe të përulur si bazë. Ata duhet të jenë të lartë dhe më i ulëti është themeli. Prandaj, princat dhe mbretërit janë të vetmuar dhe të pamëshirshëm. Kjo është rrënja e të përulurit dhe të gabuarit. Prandaj, numri i automjeteve nuk është automjet. Prandaj, ata nuk duan të jenë të pasur, si nefriti, si guri.

Lëvizja e Tao-s është dobësi, që është përdorimi i Tao-s. Parajsa

Kthimi është lëvizja e Tao-s, që është përdorimi i Tao-s. Të gjitha gjërat në botë lindin nga ekzistenca, nga mosekzistenca

Rruga është e Mirë

Rruga më e mirë është si një jorgan, dhe njeriu i mesëm mund ta praktikojë atë. Kur dëgjohet Rruga, duket se ekziston dhe zhduket. Njeriu inferior e dëgjon Rrugën dhe qesh. Ai nuk qesh dhe mendon se është Rruga. Prandaj, ekziston një shprehje: "Të kuptosh Rrugën është si përpjekje. Të përparosh është si tërheqje. Rruga është si një specie. Virtyti më i mirë është si larja. E bardha e madhe është si poshtërimi. Të zgjerosh virtytin është si pamjaftueshmëria. Të ndërtosh virtytin është si një cilësi. Sheshi i madh është pa figurë. Ena e madhe është e lirë nga suksesi. Tingulli i madh është i rrallë. Fenomenet qiellore nuk kanë ndëshkim. Rruga lëvdon dhe nuk ka emër. Vetëm Rruga është e mirë për të filluar dhe e mirë për të përfunduar.

Qi-ja e mesme përdoret për të krijuar harmoni. Ajo që bota urren janë vetëm të vetmuarit, të vejushit dhe të padenjët. Mbreti dhe duka e përdorin atë për të bërë emër për veten e tyre. Mos e zvogëloni. Nëse e zvogëloni, njerëzit e mësojnë dhe e diskutojnë atë natën dhe u mësojnë të tjerëve. Prandaj, të fortët dhe të mirët nuk duhet të vdesin.

Unë mendoj se është babai im.

Rruga lind një lind dy lind tre lind për të krijuar harmoni. Ajo ndaj së cilës njerëzit janë inferiorë janë vetëm të vetmuarit, të vejushit dhe të padenjët. Mbreti dhe duka e përdorin atë për të bërë emër për veten e tyre Unë do të përfitoj prej tij me babanë tim

Më i buti në botë galopon kundër më të fortit në botë, dhe nuk ka hyrje në boshllëk. Pesë, pra, të dish përfitimin e mosveprimit nuk është mësimdhënie, dhe përfitimi i mosveprimit është i rrallë për t'u arritur

Më galopi në botë është bota pa kohë të lirë, kështu që unë nuk do ta bëj

Cila është më afër, emri apo trupi? Cila është më shumë, trupi apo paratë? Cila është më shumë, fitimi apo humbja? Humbja është më serioze. Prandaj, të dish mjaftueshëm nuk është poshtëruese, të dish kur të ndalesh nuk është e rrezikshme, dhe mund të zgjasësh gjatë

Emri dhe

Suksesi i madh është si mungesa, dhe Qi nuk përdoret. Teprica e madhe është si një nxitim, dhe Qi është i pafund. Drejtësia e madhe është si përkulja. Aftësia e madhe është si ngathtësia. Fitorja e madhe është si shkëlqimi. Është më mirë se të ftohtit dhe të ndritshme. Ju lutem jini të bukur, dhe mund të jeni drejtësia e botës

Mjaftueshëm është si nxitimi, Qi është i zgjuar si gërmimi, dhe unë jam ftohtë

Bota ka një rrugë, dhe kuajt vrapojnë drejt plehut. Bota nuk ka rrugë, dhe kuajt e luftës lindin në periferi. Asnjë krim nuk është më i madh se dëshira, asnjë fatkeqësi nuk është më e madhe se mosdija si të ngopesh, asnjë faj nuk është më i madh se dëshira për t'u ngopur.

Kur rruga përmbyset, plehu i kuajve ikën, dhe kur nuk ka rrugë, kuajt e luftës lindin në periferi. Asnjë krim nuk është më i madh se dëshira, dhe fatkeqësia është e mjaftueshme.

Mund ta njohësh botën pa dalë nga dera, dhe mund ta njohësh rrugën e parajsës pa shikuar nga dritarja. Nëse del nga dera, mund ta

njohësh botën. Nëse nuk shikon nga dritarja, mund ta njohësh rrugën e parajsës. Nëse del nga dera, mund ta njohësh botën. Nëse nuk shikon nga dritarja, mund ta njohësh rrugën e parajsës. Nëse del nga dera, mund ta njohësh rrugën e parajsës. Nëse del nga dera, mund ta njohësh rrugën e parajsës. Nëse del nga dera, mund ta njohësh rrugën e parajsës. Nëse nuk di si të jesh i kënaqur, mund të dish si të jesh i kënaqur.

Të marrësh botën është konstante.

Për studiuesit, ata që mësojnë për mënyrën se si ditë pas dite thonë se kanë thënë se nuk do ta marrin botën. Nëse kanë thënë se nuk do ta marrin botën, nuk do ta marrin botën. Nëse kanë thënë se do ta marrin botën.

Ata që bëjnë mirë me zemrat e tyre janë të mirë, dhe ata që nuk janë të mirë janë gjithashtu të mirë. Ata janë në botë. Të gjithë janë të ndershëm. Veshët dhe sytë kolliten nga shenjtorët.

Njerëzit gjithmonë nuk kanë zemër dhe marrin zemrën e njëqind reflektimeve si zemrën e tyre. E mira është e mirë. Ata që besojnë e besojnë, dhe ata që nuk besojnë gjithashtu e besojnë. Virtyti dhe besimi janë shenjtorët në botë. E gjithë jeta është e përqendruar në Qi.

Ka trembëdhjetë njerëz në jetë, dhe njerëzit e njerëzve janë të gjithë në të vdekur. Pse ndodh kjo? Sepse Qi lind? Thuhet se ata që mbajnë jetën nuk qëllohen kur ecin në tokë. Tigrat nuk goditen nga armaturat dhe armët. Nuk ka. Nuk ka vend për Qi-në të qëllojë. Tigrat nuk kanë vend për të qëlluar. Qi-në nuk ka vend për të qëlluar. Pse ndodh kjo? Sepse Qi-në nuk ka vend për të vdekur?

Trembëdhjetë njerëz në jetë lindin dhe vdesin. Dhe njerëzit e njerëzve janë të gjithë në të vdekur. Pse ndodh kjo? Sepse Qi-në e kanë lindur? Thuhet se ata që e mbajnë mirë jetën nuk qëllohen kur ecin në tokë. Tigrat nuk goditen nga armaturat dhe armët. Nuk ka vend për Qi-në të qëllojë. Tigrat nuk kanë vend për të qëlluar. Qi-në nuk ka vend për të qëlluar. Pse ndodh kjo? Sepse Qi-në nuk ka vend

për të vdekur?

.